

INSTALACION DE POSTGRE, PHP Y APACHE EN GNU/LINUX

CESAR LOPEZ BERROCAL

**ING. HAROL BULA HERAZO
DESARROLLO WEB**

**FACULTAD DE CIENCIAS BÁSICAS E INGENIERÍA – DEPARTAMENTO DE INGENIERÍA
DE SISTEMAS Y TELECOMUNICACIONES
UNIVERSIDAD DE CÓRDOBA**



MONTERÍA, FEBRERO DE 2011

INSTALACION DE POSTGRE, PHP Y APACHE EN GNU/LINUX POR COMANDOS

PALABRAS CLAVES.

APACHE.

(Acrónimo de "a patchy server"). Servidor web de distribución libre y de código abierto, siendo el más popular del mundo desde abril de 1996, con una penetración actual del 50% del total de servidores web del mundo (agosto de 2007) La principal competencia de Apache es el IIS (Microsoft Internet Information Services) de Microsoft. Apache fue la primera alternativa viable para el servidor web de Netscape Communications, actualmente conocido como Sun Java System Web Server. Apache es desarrollado y mantenido por una comunidad abierta de desarrolladores bajo el auspicio de la Apache Software Foundation. La aplicación permite ejecutarse en múltiples sistemas operativos como Windows, Novell NetWare, Mac OS X y los sistemas basados en Unix.

Características de Apache

- * Soporte para los lenguajes perl, python, tcl y PHP.
- * Módulos de autenticación: mod_access, mod_auth y mod_digest.
- * Soporte para SSL y TLS.
- * Permite la configuración de mensajes de errores personalizados y negociación de contenido.
- * Permite autenticación de base de datos basada en SGBD.

Uso de Apache

Apache es principalmente usado para servir páginas web estáticas y dinámicas en la WWW. Apache es el servidor web del popular sistema XAMP, junto con MySQL y los lenguajes de programación PHP/Perl/Python. La "X" puede ser la inicial de cualquier sistema operativo, si es Windows: WAMP, si es el Linux: LAMP, etc.

PHP. PHP (acrónimo de *PHP: Hypertext Preprocessor*) es un lenguaje de código abierto muy popular especialmente adecuado para desarrollo web y que puede ser incrustado en HTML.

CARACTERÍSTICAS DE PHP.

Al ser un lenguaje libre dispone de una gran cantidad de características que lo convierten en la herramienta ideal para la creación de páginas web dinámicas:

- Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otras.
- Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) hasta analizar código XML.
- Ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación.
- Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.
- Soportado por una gran comunidad de desarrolladores, como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.
- Con PHP se puede hacer cualquier cosa que podemos realizar con un script CGI, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y

páginas dinámicas.

Un sitio con páginas webs dinámicas es aquel que permite interactuar con el visitante, de modo que cada usuario que visita la página vea la información modificada para propósitos particulares.

SEGURIDAD EN PHP

PHP es un potente lenguaje y el intérprete, tanto incluido en el servidor Web como módulo o ejecutado como un binario CGI, puede acceder a ficheros, ejecutar comandos y abrir comunicaciones de red en el servidor. Todas estas características hacen que lo que se ejecute en el servidor Web sea seguro por defecto.

PHP ha sido diseñado específicamente para ser un lenguaje más seguro para escribir programas CGI, Perl o C y con la correcta selección de las opciones de configuración de tiempo de compilación y ejecución se consigue la exacta combinación de libertad y seguridad que se necesita. Ya que existen diferentes modos de utilizar PHP, existe también una multitud de opciones de configuración que permiten controlar su funcionamiento. Una gran selección de opciones garantiza que se pueda usar PHP para diferentes aplicaciones, pero también significa que existen combinaciones de estas opciones y configuraciones del servidor que producen instalaciones inseguras.

RESUMEN

PHP es un lenguaje script procesado en el lado del servidor. El PHP se muestra como código embebido dentro de una página HTML. El modo de operación del PHP es el siguiente:

- El Navegador realiza una petición al servidor (se escribe la URL).
- Después el servidor ejecuta el código PHP solicitado y retorna el código HTML generado al Navegador.
- Por último el Navegador muestra la respuesta del servidor.

Este tipo de iteración permite algunas operaciones complejas como conexiones a bases de datos o ejecución de complejos programas. PHP además de soportar un numero masivo de bases de datos, incluyendo INFORMIX, ORACLE, Sybase, Solid y PostgreSQL, etc. También nos ofrece una gran variedad de funciones que nos permiten desarrollar múltiples funcionalidades que van desde enviar un e-mail, subir un archivo (upload), crear una imagen en tiempo de ejecución, interactuar con diversos protocolos de comunicación, interactuar con documentos XML, autenticación, creación dinámica de documentos PDF, entre muchas otras cosas.

Las principales características de PHP son: su rapidez; su facilidad de aprendizaje; su soporte multiplataforma tanto de diversos Sistemas Operativos, como servidores HTTP y de bases de datos; y el hecho de que se distribuye de forma gratuita bajo una licencia abierta.

POSTGRES.

PostgreSQL es un potente motor de bases de datos, que tiene prestaciones y funcionalidades equivalentes a muchos gestores de bases de datos comerciales. Es más completo que [MySQL](#) ya que permite métodos almacenados, restricciones de integridad, vistas, etc. aunque en las últimas versiones de [MySQL](#) se han hecho grandes avances en ese sentido.

Paso I.

Instalación del servidor Apache, abrimos una terminal y escribimos lo siguiente.

Paso II

Instalación de php5 y su módulo para apache2

- `$sudo apt-get install apache2`
- `sudo apt-get install php5-common php5 libapache2-mod-php5`
o también así:
- `$sudo apt-get install php5 libapache2-mod-php5`

Paso III

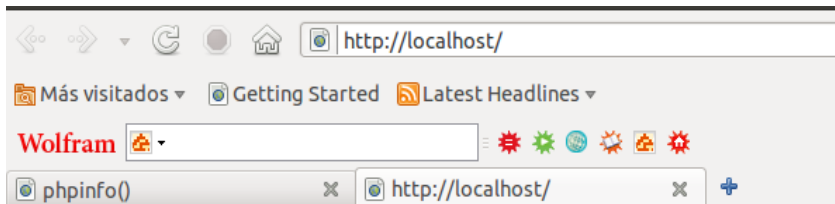
Módulo Para utilizar PostgreSQL con PHP5

- `sudo apt-get install php5-pgsql`

Paso IV

Ahora se digita la dirección del localhost de tu equipo algo así: <http://localhost/>

Lo cual debe mostrarse en el navegador de tu preferencia un mensaje en inglés algo como este [it is work!!!](#)



It works!

This is the default web page for this server.

The web server software is running but no content has been added, yet.

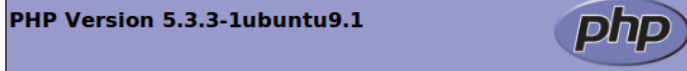
Paso V

Para que los cambios tengan efecto debes reiniciar apache.

- `sudo /etc/init.d/apache2 restart`

Para revisar que todo este funcionando correctamente escribimos el siguiente codigo en php.

```
<?
phpinfo();
?>
```



System	Linux joseph-PC 2.6.35-22-generic #35-Ubuntu SMP Sat Oct 16 20:36:48 UTC 2010 i686
Build Date	Oct 15 2010 14:06:44
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files parsed	/etc/php5/apache2/conf.d/gd.ini, /etc/php5/apache2/conf.d/mcrypt.ini, /etc/php5/apache2/conf.d/mysql.ini, /etc/php5/apache2/conf.d/mysqli.ini, /etc/php5/apache2/conf.d/pdo.ini, /etc/php5/apache2/conf.d/pdo_mysql.ini, /etc/php5/apache2/conf.d/pdo_pgsql.ini, /etc/php5/apache2/conf.d/pgsql.ini
PHP API	20090626
PHP Extension	20090626
Zend Extension	220090626
Zend Extension Build	API220090626.NTS
PHP Extension Build	API20090626.NTS

Como podemos darnos cuenta la librería de conexión de php con postgres esta habilitada.

pgsql

PostgreSQL Support	enabled
PostgreSQL(libpq) Version	8.4.5
Multibyte character support	enabled
SSL support	enabled
Active Persistent Links	0
Active Links	0

Directive	Local Value	Master Value
pgsql.allow_persistent	On	On
pgsql.auto_reset_persistent	Off	Off
pgsql.ignore_notice	Off	Off
pgsql.log_notice	Off	Off
pgsql.max_links	Unlimited	Unlimited
pgsql.max_persistent	Unlimited	Unlimited

Cambiar el puerto de escucha del servidor. Primero que todo nos ubicamos en el archico que contiene el puerto donde escucha el servidor. En nuestro caso es el puerto 80 el cual es por defecto. Que se encarga de ofrecer el contenido web (www). Nota si cambiamos el puerto vamos a tener que abrir un puerto en nuestro Router para poder operar desde Internet con nuestro servidor. Con un editor procedemos a cambiar el puerto listen 80 por listen xx.

- nano /etc/apache2/ports.conf

```
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default
# This is also true if you have upgraded from before 2.2.9-3 (i.e. from
# Debian etch). See /usr/share/doc/apache2.2-common/NEWS.Debian.gz and
# README.Debian.gz
```

```
NameVirtualHost *:80
Listen 80
```

```
<IfModule mod_ssl.c>
# If you add NameVirtualHost *:443 here, you will also have to change
# the VirtualHost statement in /etc/apache2/sites-available/default-ssl
# to <VirtualHost *:443>
```

```
# Server Name Indication for SSL named virtual hosts is currently not
# supported by MSIE on Windows XP.
```

```
Listen 443
```

```
</IfModule>
```

```
<IfModule mod_gnutls.c>
```

```
Listen 443
```

```
</IfModule>
```

```
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default
# This is also true if you have upgraded from before 2.2.9-3 (i.e. from
# Debian etch). See /usr/share/doc/apache2.2-common/NEWS.Debian.gz and
# README.Debian.gz

NameVirtualHost *:80
Listen 80

<IfModule mod_ssl.c>
# If you add NameVirtualHost *:443 here, you will also have to change
# the VirtualHost statement in /etc/apache2/sites-available/default-ssl
# to <VirtualHost *:443>
# Server Name Indication for SSL named virtual hosts is currently not
# supported by MSIE on Windows XP.
Listen 443
</IfModule>

<IfModule mod_gnutls.c>
Listen 443
</IfModule>
```

En las imágenes anteriores explicamos de manera sencilla y concisa cómo crearlos en nuestro servidor local y configurar nuestro sistema operativo para que acepte dichos nombres de host. Muchos de nosotros estamos acostumbrados a trabajar en nuestro servidor web local para nuestros proyectos en modo de prueba, y por lo general tenemos más de un proyecto en camino, sea laboral (p.e. el web site de un cliente) o personal (p.e. nuestro blog o comunidad virtual). Solemos guardar estos proyectos en carpetas separadas dentro un directorio tal como `etc/www/` que sería nuestro directorio raíz web (desde el cual llamamos a nuestro clásico `http://localhost`), y llamar a nuestros proyectos de la siguiente manera en nuestro navegador preferido:

`http://localhost/unicor`,

`http://localhost/lisblog`,

`http://localhost/casfi`, entre otros por citar un ejemplo.

En cierto modo podemos trabajar de manera regularmente cómoda, pero quizá desearíamos usar el nombre del dominio real con el que trabajamos estos proyectos en la Internet. Así podríamos hacer usar lo siguiente:

`http://www.unicor.com` en lugar de `http://localhost/unicor`, o

`http://www.linux3.com` en lugar de `http://localhost/linux3`

Y cómo hacemos esto?, pues la respuesta es sencilla, configurando la directiva VirtualHost de nuestro servidor Apache y el archivo host de nuestro Sistema Operativo Linux.

Que es un Host Virtual (ó Virtual Host) basado en Nombre ?.

No es más que hospedar múltiples dominios del estilo (www.ejemplo.com, www.otrodominio.com, etc) bajo una misma dirección IP. Este tipo de configuración es muy común en los Hosting compartidos, donde múltiples clientes comparten un mismo servidor.

El hosting virtual se tiene dos tipos diferentes:

-Hosting Virtual Basado en Direcciones IP

-Hosting Virtual Basado en Nombres

El Hosting Virtual Basado en Direcciones IP es el que usa las direcciones IP de la conexión para determinar que host virtual es el que tiene que servir, por lo que se deberá tener diferentes direcciones IP para cada host virtual, además de que dependiendo de la dirección

IP de la que se haga la petición, será el host virtual entregado, por lo que se deberá tener muchos registros para que cada host virtual sea servido, en la actualidad es muy poco utilizado ya que solo los navegadores muy antiguos son los que no soportan Hosting Virtual Basado en Nombres.

El Hosting Virtual Basado en Nombres es el que utiliza las cabeceras http enviadas por el navegador del cliente, con lo que muchos sitios web pueden compartir una sola dirección IP mediante esta técnica, ya que al usarse el nombre de dominio solicitado por el cliente, es determinado el host virtual que se debe entregar. Se recomienda usar este tipo de Hosting Virtual a menos que haya alguna razón en especial para usar el contrario, entre las que podemos encontrar:

- El hosting virtual no se puede usar con el protocolo SSL debido a la naturaleza del mismo .

-Algunos sistemas operativos y elementos de red implementan servicios de gestión de ancho de banda que no diferencian entre hosts si no se encuentran en diferentes direcciones IP.

Manos a la obra es muy sencillo. **Paso I**

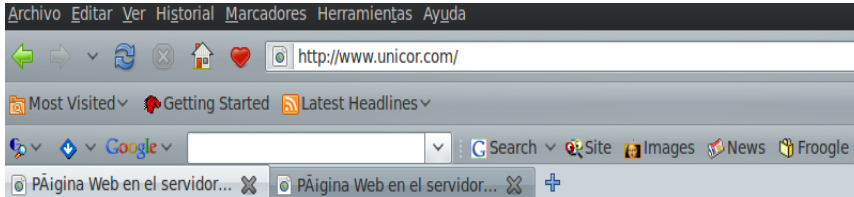
Lo que haremos ahora es colocar tantas entradas como deseemos para nuestros proyectos, relacionando la IP local con nuestros nombres de dominio. Para el ejemplo tomaremos las rutas que di anteriormente, las cuales se colocarán debajo del par 127.0.0.1 localhost:

```
GNU nano 2.2.2          Fichero: hosts          Modificado
127.0.0.1      localhost
127.0.1.1      joseph-laptop
#esta es la pagina de mi unicolor prueba
127.0.0.1      www.unicor.com
#esta es otra pagina de prueba
127.0.0.1      www.linux3.com

196.168.20.1   sistemas.com
nameserver    sistemas.com
```

Nótese que he colocado el caracter “#” . Este carácter nos permite ingresar un texto a manera de comentario. Bueno, una vez hechos estos cambios procedemos a guardar el archivo.

Para confirmar su funcionamiento abra su navegador y digite cualquiera de los nombres de host que ha agregado, recuerde de respetar el nombre completo, ya que una entrada del tipo unicor.com no tendría efecto por carecer del www. que lo antecede.



Esta es la pagina Web principal del sitio web



Esta es la pagina Web principal del sitio web

Si al ingresar el nombre del host le muestra el contenido de su <http://localhost> significa que la asignación de IP para el nombre de host a tenido efecto. Nótese que aún su nombre de host no apunta a la carpeta en la que se guarda su proyecto. Este punto lo veremos a continuación.

Paso II

Configurando el archivo httpd.conf del Apache

Ahora configuraremos nuestro servidor Apache para que acepte los nombres de dominio agregados al archivo host y los relacione con las carpetas que albergan los archivos de nuestros proyectos. Para ello abriremos el archivo **conf/httpd.conf** y nos dirigiremos a la siguiente sección:

```
#  
# Virtual Hosts  
#  
# If you want to maintain multiple domains/hostnames on your  
# machine you can setup VirtualHost containers for them. Most configurations  
# use only name-based virtual hosts so the server doesn't need to worry about  
# IP addresses. This is indicated by the asterisks in the directives below.  
#
```

Notas:

- Haga una copia de respaldo de sus archivos de configuración antes de realizar cualquier cambio en ellos.
- Si no encuentra la sección Virtual Hosts en el archivo httpd.conf quizá lo pueda ubicar en la siguiente ruta **conf/extra/httpd-vhosts.conf**

Luego procederemos a quitar el comentario de `##NameVirtualHost *:80` dejándolo de la siguiente manera:

```
#  
# Use name-based virtual hosting.  
#  
NameVirtualHost *:80
```

Esta acción nos permitira trabajar con host virtuales basados en nombres. El `*` representa un número IP y puede ser reemplazado por uno de ellos si desee. Para el ejemplo déjelo tal y como esta.

Enseguida procederemos a configurar los host virtuales, agregándolos de la siguiente manera:

Como se puede aprecia cada Virtual Host esta encerrado entre las etiquetas `<VirtualHost>` y `</VirtualHost>` que contiene el parámetro `*:80` que es `NameHostVirtual` al cual se refiere. Dentro de estas etiquetas, 2 valores son esenciales para que funcionen correctamente nuestros host virtuales:

```
GNU nano 2.2.2          Fichero: httpd.conf  
NameVirtualHost *:80  
  
#localhost  
<VirtualHost *:80>  
DocumentRoot "/var/www"  
ServerName localhost:80  
</VirtualHost>  
  
#linux lo mejor  
<VirtualHost *:80>  
DocumentRoot "/var/www/linux3"  
ServerName www.linux3.com  
</VirtualHost>  
  
#Mi U  
<VirtualHost *:80>  
DocumentRoot "/var/www/unicor"  
ServerName www.unicor.com  
</VirtualHost>
```

- **DocumentRoot.** Que es la ruta en la cual se guarda nuestra página web.
- **ServerName.** Que es el nombre del host que previamente agregamos al archivo host de nuestro S.O. Adicionalmente podemos agregar otros valores, tal como si cada host virtual fuera un host independiente (p.e. `ServerAdmin`).

Nótese algo muy importante. Si bien `localhost` es nuestro servidor local por defecto, debemos también agregarlo con un virtual host para que todo trabaje en orden, y debemos agregarle los mismos valores del `DocumentRoot` y `ServerName` que posee en su configuración global, de lo contrario no cargará el host.

Finalmente lo que haremos es para el servicio de nuestro servidor Apache y volverlo a inciar. Si todo esta correctamente configurado podemos cargar en nuestro navegador uno de los nombres de host y ver los resultados (p.e. www.unicor.com). Si carga el contenido de la carpeta que le hemos asignado en el `<VirtualHost>` y no el directorio raiz del `localhost` quiere decir que todo esta correcto. Ya tenemos configurado nuestro host virtual.

Nota: EL error mostrado en el reinicio del servicio se encuentra fuera de los alcances de este material lo cual se debe ya que tengo muchos host virtual para cuentas de usuario del sistemas.

```
root@joseph-laptop:/etc/apache2# /etc/init.d/apache2 restart
* Restarting web server apache2
apache2: Could not reliably determine the server's fully qualified domain name,
using 127.0.1.1 for ServerName
[Mon May 31 17:28:53 2010] [warn] NameVirtualHost *:80 has no VirtualHosts
... waiting .apache2: Could not reliably determine the server's fully qualified
domain name, using 127.0.1.1 for ServerName
[Mon May 31 17:28:55 2010] [warn] NameVirtualHost *:80 has no VirtualHosts
[ OK ]
root@joseph-laptop:/etc/apache2# █
```